Департамент образования и науки города Москвы
Государственное автономное образовательное учреждение
высшего образования города Москвы
«Московский городской педагогический университет»
Институт цифрового образования
Департамент информатики, управления и технологий

ДИСЦИПЛИНА: «Распределенные системы»

Отчет по практической работе №7
Тема:
«Парадигма Map Reduce»

Выполнил: Лыгин М. В.
группа: ТП-191

Москва
2022

Часть – 1. Работа с данными книги.

Работа с данными преподавателя

```
[1] !rm 1.txt

rm: cannot remove '1.txt': No such file or directory
```

```
!ls

sample_data
```

+ Код + Текст

```
[3] from google.colab import files
uploaded = files.upload()
```

Выбрать файлы 1.txt

- **1.txt**(text/plain) - 336188 bytes, last modified: 13.11.2022 - 100% done

Saving 1.txt to 1.txt

```
#объединение  лекций преподавателя в один массив данных

first = True
with open('tech.txt', 'wt', encoding='utf8') as out:
  all_data = ''.join([open('{}.txt'.format(i), encoding='utf8').read() for i in range(1, 2)])
  for line in all_data.split('\n'):
    print(line)
    if 'next' in line:
                        if not first:
                           line = line.replace('next', '')
                        else:
                           first = False
    out.write(line + '\n')
```

```
Догадка подтвердилась. Уже продав акции, я из газет узнал, что их уверенный рост был вызван секретными переговорами о слиянии. После стало известно, что некая компания пр

Этот опыт больше, чем какой-либо другой, убедил меня, что чисто технический подход к рынку работает. То есть положительный результат возможен, если учитывать только движе

Из такого расчета я и стал действовать. Я сосредоточился на изучении цен и объемов и старался не обращать внимания на слухи, подсказки и фундаментальные данные. Я принял
```

```
Инвестор-танцор. Как я заработал 2 миллиона долларов на фондовом рынке
Николас Дарвас

Оперировавший на рынке в 1950-1960-е годы, Дарвас ныне признан пионером технического анализа и одним из лучших трейдеров второй половины XX века. Танцор по профессии, он разработал уникальный метод выбора акций именно для непрофесс



Николас Дарвас

Инвестор-танцор. Как я заработал 2 миллиона долларов на фондовом рынке



```

```
import requests
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import re
import string
from collections import defaultdict
import nltk
from nltk.corpus import stopwords
import string
# Lowering the case, removing punctuations and numbers

text_clean = text.replace('-', ' ') #replacing hyphens with whitespace
text_clean = text_clean.replace('.', ' ') #replacing hyphens with whitespace
text_clean = text_clean.replace('’', '') #replacing hyphens with whitespace
text_clean = text_clean.replace(')', '') #replacing hyphens with whitespace
text_clean = text_clean.replace('(', '') #replacing hyphens with whitespace
text_clean = text_clean.replace('', '') #replacing hyphens with whitespace
text_clean = text_clean.replace('\n', ' ') #replacing hyphens with whitespace
text_clean = text_clean.replace('\x92', '') #replacing hyphens with whitespace
text_clean = text_clean.replace('\x94', '') #replacing hyphens with whitespace
text_clean = text_clean.replace('\x86', '') #replacing hyphens with whitespace
text_clean = text_clean.replace('№', '') #replacing hyphens with whitespace \t « \xa0
#text_clean = text_clean.replace('\xa0', ' ') #replacing hyphens with whitespace
text_clean = text_clean.replace('–', ' ') #replacing hyphens with whitespace
text_clean = text_clean.replace('\t', ' ') #replacing hyphens with whitespace
text_clean = text_clean.replace('«', '') #replacing hyphens with whitespace
text_clean = text_clean.replace('»', '') #replacing hyphens with whitespace —
text_clean = text_clean.replace('—', ' ') #replacing hyphens with whitespace
text_clean = text_clean.replace('/', ' ') #replacing hyphens with whitespace
table=str.maketrans('','',string.punctuation)
text_clean = text_clean.translate(table)

#removing numbers
text_clean = re.sub(r'\d', '', text_clean)

text_lower = text_clean.lower() #lowercasing
#len(text_lower)
```

```
#len(text_lower2)
text_lower3
```

```
'',
 '',
'о',
'том',
'что',
'это',
'был',
'человек',
'незаурядного',
'таланта',
'свидетельствует',
'тот',
'факт',
'что',
'дарвас',
'и',
'его',
'партнерша',
'',
'',
 '',
'родная',
'сестра',
'',
'',
 '',
'стали',
'самой',
'высокооплачиваемой',
'танцевальной',
'парой',
'своего',
'времени',
'',
 '',
'упоминание',
'в',
'данной',
'книге',
'контракта',
'со',
'знаменитым',
'ночным',
'клубом',
```

```
text_test = ' '.join([ch for ch in text_lower3 if ch not in spec_chars])
text_test
```

```
'инвестор танцор как я заработал миллиона долларов на фондовом рынке николас дарвас оперировавший на рынке в е годы дарвас ныне признан пионером технического анализа и одним из лучших трейдеров второй половины xx века танцор по профе ссии он разработал уникальный метод выбора акций именно для непрофессионалов не требовавший глубокого погружения в состояние дел компаний книга содержит подробный рассказ о методе и истории личного успеха дарваса впервые изданная в с ша в году она стала супербестселлером и выдержала многочисленные переиздания по ней до сих пор учатся новые поколения инвесторов предназначена для всех кто интересуется инвестициями в акции и думает о том как заработать на фондовом р ынке николас дарвас инвестор танцор как я заработал миллиона долларов на фондовом рынке все права защищены никакая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письме нного разрешения владельца авторских прав предисловие партнера издани…'
```

```
from nltk import word_tokenize
nltk.download('punkt')
text_tokens = word_tokenize(text_test)
text_tokens
```

```
 'не',
 'выдыхалось',
 'полностью',
 'в',
 'результате',
 'дарвасу',
 'удалось',
 'избежать',
 'стандартной',
 'ошибки',
 'спекулянтов',
 'новичков',
 'которые',
 'любят',
 'продавать',
 'акции',
 'показавшие',
 'прибыль',
 'и',
 'держать',
 'акции',
 'показывающие',
 'убытки',
 'то',
 'есть',
 'ждать',
 'когда',
 'эти',
 'акции',
 'вырастут',
 'при',
 'этом',
 'они',
 'исходят',
 'из',
 'расхожего',
 'убеждения',
```

```
dftt = pd.DataFrame(text_tokens, columns=['Words'])
dftt = dftt.dropna(subset=['Words'])
sentences=dftt['Words'].values.tolist()
len(sentences)
```

```
28808
```

```
normal_text
```

```
'не',
'выдыхаться',
'полностью',
'в',
'результат',
'дарваса',
'удаться',
'избежать',
'стандартный',
'ошибка',
'спекулянт',
'новичок',
'который',
'любить',
'продавать',
'акция',
'показать',
'прибыль',
'и',
'держать',
'акция',
'показывать',
'убыток',
'то',
'есть',
'ждать',
'когда',
'этот',
'акция',
'вырасти',
'при',
'это',
'они',
'исходить',
'из',
'расхожий',
```

```
[18] shortest_word_len = 2
     result = [s for s in normal_text if len(s) > shortest_word_len]
     text_lower2=result
     map11 = []
     map22 = []
     #a_to_m = ['a','b','c','d','e','f','g','h','i','j','k','l','m','n', 'o','p','q','r','s','t','u','v','w','x','y','z']
     a_to_m = ['а', 'б', 'в', 'г', 'д', 'е', 'ж', 'з', 'и', 'й', 'к', 'л', 'м', 'н', 'о', 'п', 'р', 'с', 'т', 'у', 'ф', 'х', 'ц', 'ч', 'ш', 'щ', 'ъ', 'ы', 'ь', 'э', 'ю', 'я', 'ё']
     for words in text_lower2:
         if words[0][0] in a_to_m:
             map11.append(words)
         else:
             map22.append(words)
     text_lower2=map11
     text_lower=" ".join(map(str, text_lower2))
     text_lower
```

```
рнуться ный йорк арена мой бесславный деяние где быть расстояние короткий поездка такси биржа инструкция остаться тот между уолл стрит должный остаться тот тысяча миля пусть брокер слать телеграмма как если быть гонконг карачи ил
и стокгольм ещё они должный слать котировка никакой другой акция кроме тот что просить сообщать какой новый бумага поскольку это уже быть разряд слух сам быть выбирать новый акция как делать всегда при чтение еженедельный финансо
вый бюллетень если увидеть интересный бумага порог рост попросить котировка большой чем один новый бумага раз потом как ранний быть внимательно изучать они прежде чем связываться тот кто выжить авиакатастрофа следовать немедленно
лететь куда снова иначе уже никогда преодолеть страх вот так видеть только один способ доказать надёжность мой метод поехать аэропорт взять билет ный йорк глава два миллион доллар возвратиться ный йорк третий неделя февраль год п
очувствовать себя совершенно оправиться удар получить время помутнение рассудок начать играть биржа новый рана который сам себя нанести глупость всё ещё напоминать себя чувствовать что идти поправка хорошенько выучить последний у
рок надо твёрдо держаться свой система стоить отойти она весь однажды полобоваться что это привести весь мой финансовый конструкция немедленно оказаться под удар едва рассыпаться как карточный домик ный йорк первый дело заняться
сооружение вокруг себя глухой стена который должный помешать повторение любой прежний ошибка для начало разделить операция между шесть брокер так кто либо быть трудный проследить для защита любой вмешательство они сторона возвест
и особый препятствие который пользоваться сей день вот что оно себя представлять распорядиться чтобы брокер слать телеграмма после закрытие торг это случай они приходить после шесть вечер как раз вставать это время таков быть рез
ультат многолетний выступление ночной клуб кроме тот течение день телефонистка быть запретить соединять кто быть такой образ всё событие уолл стрит происходить когда быть кровать пока они работать спасть они быть добраться если с
лучаться нечто непредвиденный мой делегат выступать стоп лосса семь вечер садиться работа изучать телеграмма раздумывать над будуший сделка прежде покупать вечерний газета цена закрытие уолл стрит вырывать страница сегодняшний ко
тировка всё остальной выбрасывать собираться читать никакой комментарий репортаж сколь угодный правдивый они мочь увести сторона вот потом когда уолл стрит всё уже спасть брать телеграмма газетный лист приступать дело всё тот нед
еля пока лечить раненый самолюбие два остаться бумага продолжать расти идти вверх почти безостановочно пока застопориться район мой последний визит ный йорк рост составить также держаться молодцом уже лезть выше это обещать насто
ящий жирный куш решить что трогать они незачем хорошенько укрыться свой новый стена научить горький опыт стать потихоньку оглядка прощупывать рынок вот некоторый успешный сделка тот время купить долл долл продать долл долл прибыл
```

```
#Reducing the first a-m list
list3 = reduce1(map1)

#Reducing the second n-z list
list4 = reduce2(map2)

#Merging the two reduced lists
answer_list = list3 + list4
print(answer_list)
```

```
[('инвестор', 1), ('танцор', 1), ('как', 1), ('заработать', 1), ('миллион', 1), ('доллар', 1), (
[('ать', 1), ('несколько', 1), ('раз', 1), ('как', 1), ('выражаться', 1), ('дарвас', 1), ('если'
('абсолютный', 1)
21272
0
[('абсолютный', 1), ('абсолютный', 1), ('аванс', 1), ('авантюра', 1), ('авария', 1), ('август',
[]
[('абсолютный', 2), ('аванс', 1), ('авантюра', 1), ('авария', 1), ('август', 6), ('авеню', 2), (
```

```
[23] #Reducing the first a-m list
     list3 = reduce1(map1)

     #Reducing the second n-z list
     list4 = reduce2(map2)

     #Merging the two reduced lists
     answer_list = list3 + list4
     print(answer_list)
```

```
[('абсолютный', 2), ('аванс', 1), ('авантюра', 1), ('авария', 1), ('август', 6),
```

Reducing the second n-z list

```
[24] list4 = reduce2(map2)
```

Merging the two reduced lists

```
answer_list = list3 + list4
print(answer_list)
```

```
[('абсолютный', 2), ('аванс', 1), ('авантюра', 1), ('авария', 1), ('август', 6),
```

```
df = pd.DataFrame(answer_list, columns=['Word', 'Frequency'])
print(df)
df.info()
df1=df[['Word','Frequency']].sort_values(ascending=False,by='Frequency')
df1
```

```
            Word  Frequency
0     абсолютный          2
1          аванс          1
2       авантюра          1
3         авария          1
4         август          6
...          ...        ...
4284      яснеть          1
4285        ясно          6
4286     ясность          2
4287       ясный          4
4288        ящик         68

[4289 rows x 2 columns]
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4289 entries, 0 to 4288
Data columns (total 2 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   Word       4289 non-null   object
 1   Frequency  4289 non-null   int64 
dtypes: int64(1), object(1)
memory usage: 67.1+ KB
```

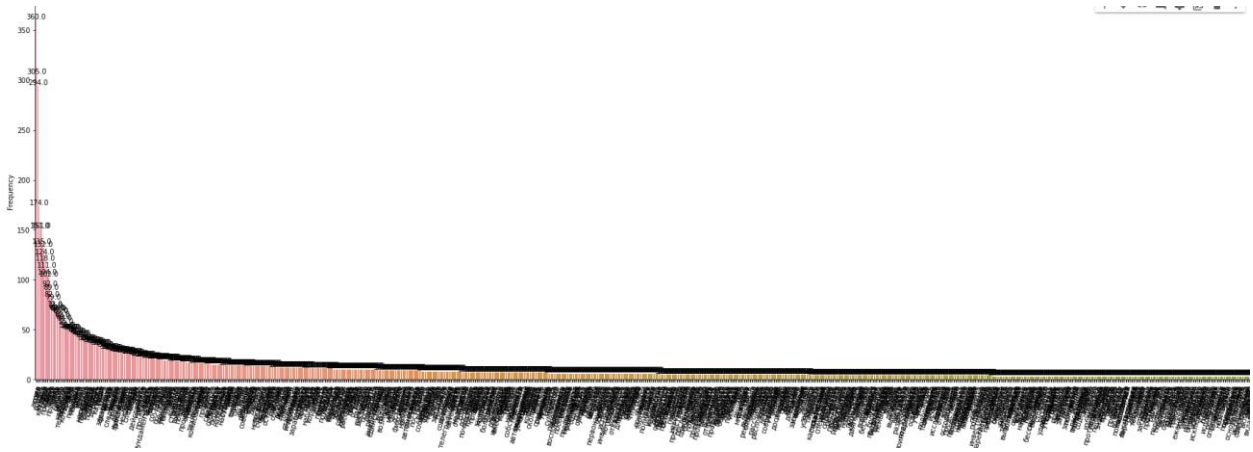
	Word	Frequency
4188	что	560
210	быть	526
2041	они	515
33	акция	360
794	долл	305
...	...	...
2265	педаль	1
2266	пелена	1
538	высвободить	1
2268	первозданный	1
464	втолковать	1

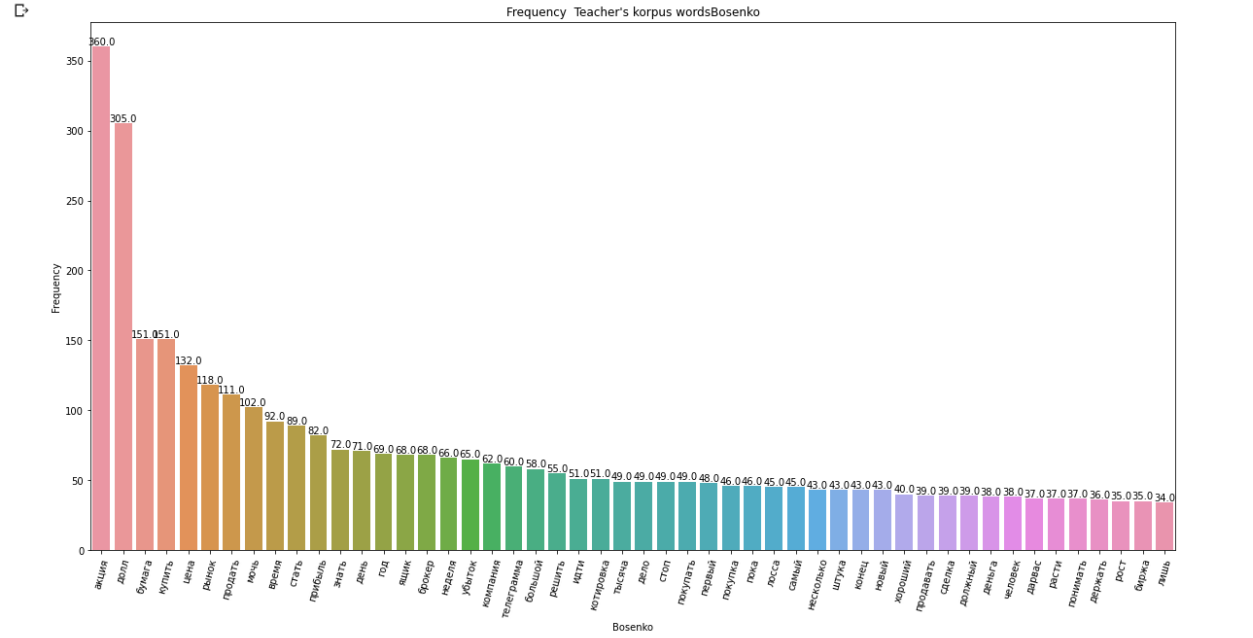
4289 rows × 2 columns

```
df_teach=df1.copy()
df_teach_clear=df_teach[~df_teach.Word.isin(df_stop.stop_ru)].reset_index(drop=True)
df_teach_clear.rename(columns = {'Word':'Bosenko'}, inplace = True)
print(df_teach_clear.sort_values(by ='Frequency', ascending = 0).reset_index(drop=True))
print(df_teach_clear.count())
df_teach_clear2=df_teach_clear.sort_values(by ='Frequency', ascending = 0).reset_index(drop=True)
```

```
          Bosenko  Frequency
0           акция        360
1            долл        305
2             это        294
3             всё        174
4          бумага        151
...           ...        ...
4207   смешивать          1
4208   смешанный          1
4209        смех          1
4210       смесь          1
4211  втолковать          1

[4212 rows x 2 columns]
Bosenko      4212
Frequency    4212
dtype: int64
```





(-0.5, 399.5, 199.5, -0.5)

Часть – 2. Сопоставление с рецензией

Final Dataframe

```
df = pd.DataFrame(answer_list, columns=['Word', 'Frequency'])
df
```

	Word	Frequency
0	автобиография	1
1	активность	2
2	акции	8
3	акций	3
4	акция	3
...	...	...
374	ящик	1
375	ящика	2
376	ящике	1
377	ящики	1
378	ящиков	1

379 rows × 2 columns

```
[26] df1=df[['Word','Frequency']].sort_values(ascending=False,by='Frequency')
     df1
```

0 сек.

	Word	Frequency
74	если	11
2	акции	8
329	том	6
361	что	6
371	является	5
...	...	...
141	лучшему	1
140	лучше	1
139	лишь	1
138	лично	1
378	ящиков	1

379 rows × 2 columns

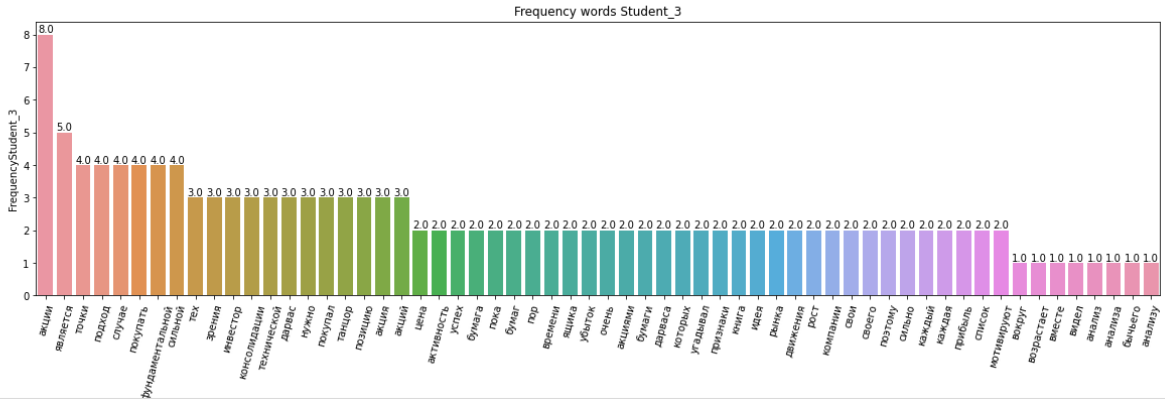
```
df_stop= pd.read_excel("stop-words-ru.xlsx")
df_stop
#print(df_teach)
print(df_stop.count())
#убираем слова-стоп
#очистка текста преподавателя применяя словарь стоп-слов используем ~
df_student=df1.copy()
df_student_clear=df_student[~df_student.Word.isin(df_stop.stop_ru)].reset_index(drop=True)
df_student_clear.rename(columns = {'Word':'WordStudent_3'}, inplace = True)
df_student_clear.rename(columns = {'Frequency':'FrequencyStudent_3'}, inplace = True)
print(df_student_clear.sort_values(by ='FrequencyStudent_3', ascending = 0).reset_index(drop=True))
print(df_student_clear.count())
df_student_clear2=df_student_clear.sort_values(by ='FrequencyStudent_3', ascending = 0).reset_index(drop=True)


#df2=df1[df1['Word'].map(len) > 3]
```

```
Unnamed: 0    161
stop_ru       161
dtype: int64
    WordStudent_3  FrequencyStudent_3
0           акции                   8
1        является                   5
2           точки                   4
3          подход                   4
4          случае                   4
..            ...                 ...
328    сильнейшую                   1
329    сильнейшей                   1
330        сделка                   1
331        теорию                   1
332        ящиков                   1

[333 rows x 2 columns]
WordStudent_3         333
FrequencyStudent_3    333
dtype: int64
```

```
#Plotting the top 5 films by revenue
#setting the figure size
plt.figure(figsize=(20,5))
#creating a bar plot
ax=sns.barplot(x='WordStudent_3',y='FrequencyStudent_3',data=df_student_clear2.head(60))
#rotating the x axis labels
ax.set_xticklabels(labels=df_student_clear2.head(60)['WordStudent_3'],rotation=75)
#setting the title
ax.set_title("Frequency words Student_3")
#setting the Y-axis labels
ax.set_ylabel("FrequencyStudent_3")
#Labelling the bars in the bar graph
for p in ax.patches:
  ax.annotate(p.get_height(),(p.get_x()+p.get_width()/2,p.get_height()),ha='center',va='bottom')
```



```
[34] !rm Analise.xlsx
```

```
rm: cannot remove 'Analise.xlsx': No such file or directory
```

```
[35] from google.colab import files
     uploaded = files.upload()
```

Выбрать файлы Analise.xlsx

- **Analise.xlsx**(application/vnd.openxmlformats-officedocument.spreadsheetml.sheet) - 90411 bytes, last modified: 15.11.2022 - 100% done

```
Saving Analise.xlsx to Analise.xlsx
```

```
df_teach = pd.read_excel("Analise.xlsx", 'teach')
df_teach
#print(df_teach)
print(df_teach.count())
```

```
Bosenko      4203
Frequency    4203
dtype: int64
```

```
[37] df_stud = pd.read_excel("Analise.xlsx", 'student_3')
     #print(df_stud)
     print(df_stud.count())
```

```
WordStudent_3         333
FrequencyStudent_3    333
dtype: int64
```

```
[38] #частота совпадений студента с корпусом данных
     df_work=df_stud.copy()
     df_student_ok=df_work[df_work.WordStudent_3.isin(df_teach.Bosenko)].reset_index(drop=True)
     df_student_ok.rename(columns = {'WordStudent_3':'WordSt_3Tch'}, inplace = True)
     df_student_ok.rename(columns = {'FrequencyStudent_3':'FrSt_3'}, inplace = True)
     print(df_student_ok.sort_values(by ='WordSt_3Tch', ascending = 0).reset_index(drop=True))
     print(df_student_ok.count())
```

```
    WordSt_3Tch  FrSt_3
0          ящик       1
1           эго       1
2         часто       1
3          цена       2
4         успех       2
..          ...     ...
81       бумага       2
82  большинство       1
83       анализ       1
84        акция       3
85   активность       2

[86 rows x 2 columns]
WordSt_3Tch    86
FrSt_3         86
dtype: int64
```

```
[39] #частота совпадений преподавателя с  словами студента- для определения частоты встречаемости слов у преподавателя, которые использовал студент
     df_work2=df_teach.copy()
     df_teach_ok=df_work2[df_work2.Bosenko.isin(df_stud.WordStudent_3)].reset_index(drop=True)
     df_teach_ok.rename(columns = {'Bosenko':'WordSt_3Tch'}, inplace = True)
     df_teach_ok.rename(columns = {'Frequency':'FrTch'}, inplace = True)
     print(df_teach_ok.sort_values(by ='WordSt_3Tch', ascending = 0).reset_index(drop=True))
     print(df_teach_ok.count())
```

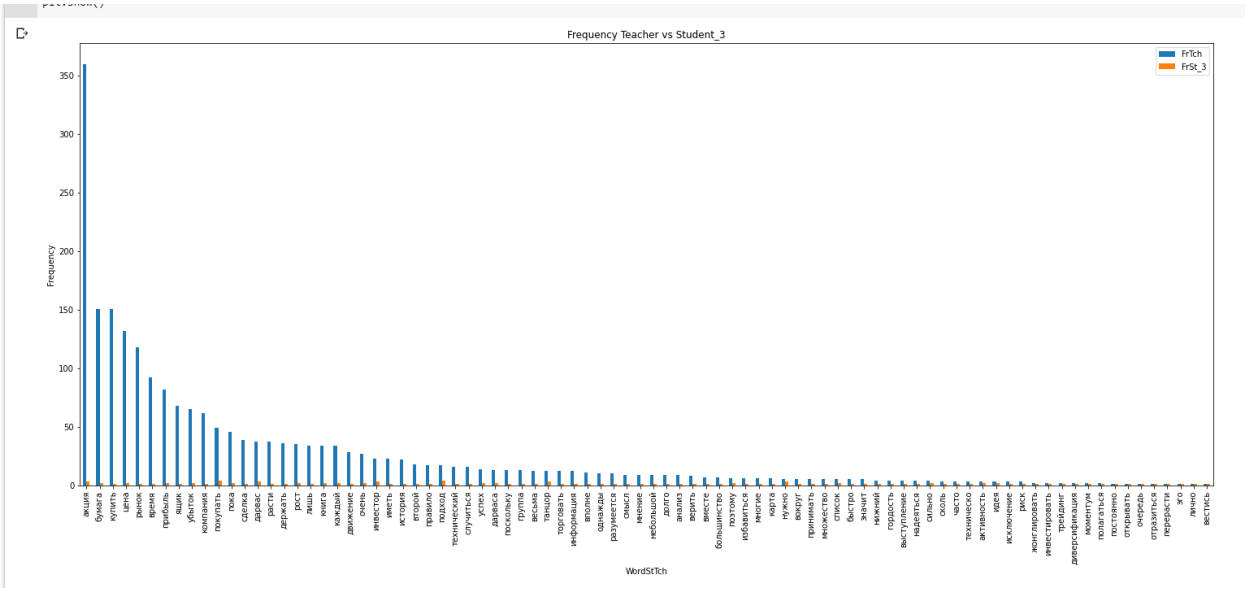
```
    WordSt_3Tch  FrTch
0          ящик     68
1           эго      1
2         часто      3
3          цена    132
4         успех     14
..          ...    ...
81       бумага    151
82  большинство      7
83       анализ      9
84        акция    360
85   активность      3

[86 rows x 2 columns]
WordSt_3Tch    86
FrTch          86
dtype: int64
```

```
#Объединяем частоты студента и преподавателя
res = df_teach_ok.merge(df_student_ok)
res
```

	WordSt_3Tch	FrTch	FrSt_3
0	акция	360	3
1	бумага	151	2
2	купить	151	1
3	цена	132	2
4	рынок	118	1
...	...	...	...
81	отразиться	1	1
82	перерасти	1	1
83	эго	1	1
84	лично	1	1
85	вестись	1	1

86 rows × 3 columns




```
res.plot(kind='bar', figsize=(30,10))
x_pos = [i for i, _ in enumerate(x)]
plt.xlabel("WordStTch",fontsize=16)
plt.ylabel("Frequency", fontsize=16)
plt.title("Book vs Review")
plt.xticks(x_pos, x,fontsize=16)
plt.show()
```